

A deficiency of many current microkernel designs is that they feature separate and computationally expensive priority inheritance mechanisms. Hence, turning them off to boost operating system performance is always going to be a temptation. An architecture which provides automatic priority inheritance without any performance penalty is therefore preferable to existing designs, since it would not force robustness and performance to be traded off against each other. This is especially true where the nanokernel is used to host one or more operating systems, since any unreliability in the nanokernel translates directly into unreliability in the operating system as a whole.

It is therefore an object of the present invention to provide an improved form of computing device which is able to display automatic priority inheritance without any performance penalty.

According to a first aspect of the present invention there is provided a computing device comprising a scheduler incorporating an algorithm for ordering the running of threads of execution having different priorities; and including a list of threads which are scheduled to run on the device, ordered by priority; the device further comprising at least one locking mechanism for blocking access to a resource of the device from all threads except for a thread that holds the locking mechanism; and in which a scheduled thread which is blocked from running causes the thread which holds the locking mechanism to run.

According to a second aspect of the invention there is provided a method of operating a computing device, the method comprising providing a scheduler incorporating an algorithm for ordering the running of threads of execution having different priorities, and including a ready list of threads which are scheduled to run on the device, ordered by priority; providing at least one locking mechanism for blocking access to a resource of the device from all threads except for a thread that holds the locking mechanism; and arranging for a scheduled thread which is blocked from running because the resource it requires is locked not to be removed from its place on the ready list and instead the thread which holds the locking mechanism is caused to run.



According to a third aspect of the present invention there is provided an operating system for a computing device, the operating system comprising a scheduler incorporating an algorithm for ordering the running of threads of execution having different priorities, and including means for providing a ready list of threads which are scheduled to run on the device, ordered by priority; at least one locking mechanism for blocking access to a resource of the device from all threads except for a thread that holds the locking mechanism; and means for arranging for a scheduled thread which is blocked from running because the resource it requires is locked not to be removed from its place on the ready list and instead the thread which holds the locking mechanism to be caused to run.

An embodiment of the present invention will now be described, by way of further example only, with reference to the accompanying drawing which illustrates examples of monolithic and micro kernel architectures.

The present invention will be described with specific reference to the Symbian OS<sup>TM</sup> operating system available from Symbian Limited of London, England. However, it is to be understood that the principles of the present invention may also be used to equal advantage in other types of operating system.

The Symbian OS operating system includes a pre-emptive multi-threaded nanokernel style layer providing hard real-time support. It is within this nanokernel that an implementation of a single mechanism which is referred to as a *fast mutex* is to be found. This combines the benefits of a method for kernel threads to provide mutual exclusion locks on resources with the benefits of a method for automatic priority inheritance for the task holding the mutex. The multi-threaded nature of the kernel makes it suitable for use in either single-processor or symmetrical multiprocessing (SMP) systems, and the fast mutex can be used in either configuration.

It is not considered necessary to fully describe the Symbian OS operating system in order to provide a sufficient understanding of this invention. Thus,



7a

the following description is restricted to those parts of the operating system relevant to the invention. The Symbian OS kernel is a hybrid between the monolithic and microkernel approaches shown in figure 1, and therefore